REMARKS/ARGUMENTS

Characteristics of Identifier-Based Encryption (IBE)

In identifier-based encryption, the data to be encrypted is encrypted using as encryption parameters:

- public data of a trusted party;
- an encryption key string – this can be any arbitrary string and does not need to be known in advance by the trusted party (the trusted party is responsible for generating the corresponding private key).

The private key is generated by the third party using:

- the encryption key string;
- private data of the trusted party, this private data being related to the public data of the trusted party.

Using the public data of the trusted party as an encryption parameter ensures that only that trusted party can generate the private key needed for decryption.

It should be noted that the public and private data of the trusted party can also be thought of as a public/private key pair – in other words, the public and private data of the trusted party are cryptographically related to each other; in the exemplary embodiment given in the specification, the private data are the primes *p, q* and the related public data is the modulus N formed as the product of *p & q*.

The Present Application

In the present application the data to be encrypted is personal data and the encryption key string comprises policy data indicative of conditions, differing from recipient identity, to be satisfied before access is given to said personal data.

Related Applications

The specification has been amended to refer to a related US Patent 7,219,226 granted May 15, 2007 and a presently pending divisional application (USSN 11/644862) that is based on the application which matured into that patent.

Claim Amendments

In the amended claim 1 the policy data is specified as being "constituted by a first form of a policy indicative of conditions to be satisfied before accessed is given to said personal data", and the encrypted data is stated as being "provided to the recipient together with a second form of said policy, the first and second forms of the policy being different from each other". This amendment has the effect that the trusted party cannot create the decryption key by using the form of the policy sent with the encrypted data by the first party.

The private data of the trusted party has also been stated to be "cryptographically" related to the public data. This is somewhat more specific than simply saying "related".

The 35 USC § 102 Rejection based on Sweet (US 20020031230)

The Sweet reference does not use identifier-based encryption, rather Sweet discloses an improved form of a key management system based on the CMK ("Constructive Key Management") standard – see paragraph [0008]. The purpose of the system is to enable an object to be encrypted by a user such that only other users that meet certain requirements (such as work in a particular department on a particular project) can decrypt the object. The requirements are specified in terms of a Boolean set of one or more credentials that a user must possess (by 'Boolean set' I mean that credentials can be combined in AND or OR relationships as explained in paragraph [0144]).

The credentials and other security parameters of a user are stored in a user security profile that can either be held by the key management system itself or in a user token such as a smart card – see paragraphs [0113]-[0117].

In more detail, and with reference to Figure 3 and paragraphs [0122]-[0129], a "working key" for encrypting an object 220 is constructed by combining three values, namely:

a Maintenance value 210,

a Domain value 205, and

a Pseudo Random value 215.

The encrypting user obtains the first two values from the user's security profile and generates the third value. The first two values will be shared by many users; the pseudo random value not only provides uniqueness to the working key but also serves as the vehicle by which the encrypting party sets the credentials required by another user to enable the latter to decrypt the object.

More particularly, the pseudo random value is stored in encrypted form in a header 235 of the encrypted object. See paragraph [0134]. The pseudo random value is encrypted using a key based on the credential set that the encrypting user requires another user to have in order to be able to access the object. The required credentials are also identified in the header 235.

A user wishing to decrypt the object 220 must possess in its security profile, not only the Maintenance and Domain values used to form the working key, but also the set of credentials used to encrypt the pseudo random value stored in the object header. Paragraph [0147] describes how a user goes about decrypting an object – either by itself accessing its security profile to extract the required data for constructing the working key, or by having the system do this and provide the working key to the user.

Comparing Claim 1 to the Sweet reference, it is believed that the Examiner makes the following associations: The "owner of personal data" of claim 1 is analogous to the position of a user of Sweet wishing to encrypt an object. The "recipient" of claim 1 is clearly a user of Sweet who is intended to be able to access the encrypted object. The "trusted party" of claim 1 is the key management server system of Sweet (excluding user machines), the system clearly needing to be trusted by all users.

Actually, Sweet provides two possible candidates for the "data" to be encrypted in claim 1, these two candidates being:

(i) the object 220 to be encrypted, and

(ii) the pseudo random value that is encrypted for inclusion in the object header.

Each candidate will be considered in turn below.

As regards data encryption, claim 1 requires this to be based on encryption parameters comprising:

- public data provided by a trusted party, and
- an encryption key string formed using at least policy data constituted by a first form of a policy indicative of conditions to be satisfied before access is given to said personal data,

Leaving aside the "public data" limitation for the moment, as for an encryption parameter using policy data, if the data to be encrypted is taken as the pseudo random value of Sweet, then the credential set used to form an encryption key for encrypting the pseudo random value can be considered to be the claim 1 "policy".

If, however, the data to be encrypted is taken to be the Sweet 'object', then the parameters of encryption do not include the credentials as these are only used to provide access to the pseudo random value. Perhaps the Examiner believes that the Domain and Maintenance values that are used in forming the working key are encryption parameters and serve as policy elements.

But amended Claim 1 next requires that the encrypted data is provided to a recipient "together with a second form of said policy, the first and second forms of the policy being different from each other".

Next we come to the role of the trusted party of claim 1 in decrypting the data. This requires the trusted party to use the encryption key string and "private data cryptographically related to said public data" to determine a decryption key. The encryption key string comprises the first form of the policy – that is, the credentials where the encrypted data is taken as the pseudo random value, or the Domain and Maintenance values where the encrypted data is taken as the Sweet object 220. As already noted, paragraph [0147] describes the Sweet system accessing a user's security profile in order to form the working key for decrypting the object of interest; en route to forming the working key the Sweet system necessarily forms a decryption key for accessing the pseudo random value in the object header. Thus whether the encrypted data is the pseudo random value or the object, the Sweet system forms an appropriate decryption key.

So, we come to an important point: does the key management system of Sweet (the "trusted party" of claim 1) use "private data cryptographically related to said public data" to determine a decryption key as required by claim 1? Is there anything in Sweet to suggest that this is done? Note that paragraph [0147] states that the system, when generating the working key, uses "information contained within the security profile, along with information contained within the encrypted data file". Neither of these types of information are "private data" of the system. The security profile information is conceptually private data of the user even when held by the system. The encrypted data file information is also not private to the system as it was known to the encrypting user.

Furthermore, where does Sweet disclose "public data" of the system used in the encrypting process and is also cryptographically related to system private data used to form the decryption key.

If the examiner is of the view that in the embodiments of Sweet where the user security profiles are held by the system, that the contents of the user security profiles are, in practice, "private data" of the system, the Applicant asserts that such a view would not be correct for the following reasons:

- although paragraph [0147] of Sweet describes the generation of a working key for decryption being done by the system, there appears to be no corresponding disclosure of the system effecting object encryption, this being described (for example in paragraph [0141]) as being done by the object's creator. Since object encryption requires access to Domain, Maintenance and credential values, these values cannot be considered as data private to the key management system (this 'system' does not include the user machines). On the other hand, if an embodiment or modification of Sweet had the object encryption being effected by the key management system (so that the Domain, Maintenance and credential values were 'private' to the system), then the claim 1 requirement that encryption is done by the owner of personal data is not satisfied.

- an interpretation that Domain, Maintenance and credential values are private data of the Sweet key management system makes it very difficult to see what data, used in the encryption process, could possibly be considered "public data" of the system (note that claim 1 requires that "public data" of the system be used as an encryption parameter).

Furthermore, because the system of Sweet must necessarily be accessed by a user wishing to decrypt an object, there is no reason to modify Sweet to employ public and private data as required by claim 1 since the purpose of such data is to force anyone wishing to access an encrypted object to use the trusted party chosen by the encrypting party (this choice being embodied in the particular public data used in the encryption process).

Sweet therefore fails to disclose all the features of amended claim 1. Similar differentiating arguments apply in respect of amended independent claim 37. Claim 53 has also been amended to more clearly differentiate that claim from Sweet.

Withdrawal of the rejections and allowance of the claims are respectfully requested.

The Commissioner is authorized to charge any additional fees which may be required or credit overpayment to deposit account no. 08-2025. In particular, if this response is not timely filed, then the Commissioner is authorized to treat this response as including a petition to extend the time period pursuant to 37 CFR 1.136 (a) requesting an extension of time of the number of months necessary to make this response timely filed and the petition fee due in connection therewith may be charged to deposit account no. 08-2025.

I hereby certify that this correspondence is being electronically transmitted to the United States Patent Office on

9 January 2008
(Date of Transmission)

Stacey Dawson
(Name of Person Transmitting)

Stacey Dawson
(Signature)

9 January 2008
(Date)

Respectfully submitted,

Richard P. Berg
Attorney for the Applicant
Reg. No. 28,145
LADAS & PARRY
5670 Wilshire Boulevard,
Suite 2100
Los Angeles, California 90036
(323) 934-2300 voice
(323) 934-0202 facsimile